International Journal of Computer and Communication Technology

Volume 4 | Issue 1 Article 9

January 2013

Ant Colony Optimization. A Computational Intelligence Technique

Bijaya Kumar Nanda
Research Scholar , Department of ICT, F.M. University, bijaykumarnanda@gmail.com

Gyanesh Das
Asst Professor., Department of ENTC, DRIEMS, gyaneshdas@gmail.com

Recommended Citation

Nanda, Bijaya Kumar and Das, Gyanesh (2013) "Ant Colony Optimization. A Computational Intelligence Technique," *International Journal of Computer and Communication Technology*: Vol. 4 : Iss. 1 , Article 9.
DOI: 10.47893/IJCCT.2013.1169
Available at: https://www.interscience.in/ijcct/vol4/iss1/9

Ant Colony Optimization.
A Computational Intelligence Technique

[1]Bijaya Kumar Nanda, [2]Gyanesh Das,
[1]Research Scholar , Department of ICT, F.M. University.
[2]Asst Professor., Department of ENTC, DRIEMS.

Abstract-Ant colony optimization (ACO) is a novel computational technique inspired by a foraging behavior of ants has been successfully applied for solving real world optimization problems. This behavioral pattern inspires artificial ants for the search of solutions to the various types of optimization problems. ACO is a probabilistic search approach founded on the idea of evolutionary process. In this paper, we present an overview of ant colony optimization and ACO variants up to now. we also summarize various types of applications. Finally we focus on some research efforts directed at receiving a dipper understanding of the ant colony optimization algorithms.

INTRODUCTION

Ant Colony Optimization (ACO) is a paradigm for designing meta heuristic algorithms for optimization problems. The first algorithm which can be classified within this framework was ant system[1] presented in 1991. Since then many diverse variants of the basic principle have been reported in the literature. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions. The characteristic of ACO algorithms is their explicit use of elements of previous solutions. The probability distribution in ACO is also explicitly defined by previously obtained solution components. The particular way of defining components and associated probabilities is problem-specific and can be designed in different ways. In general, the ACO approach attempts to solve an optimization problem by repeating the following two steps:

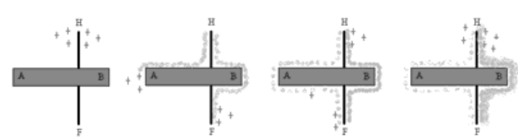
1. Using a pheromone model, that is, a parameterized Probability distribution over the solution space;
2. The candidate solutions are used to modify the pheromone values in a way that is deemed to bias future sampling toward high quality solutions.

The goal of this article is to introduce ant colony optimization and to survey the best performing ACO Variants. Section I provides some background information on the foraging behavior of ants. Section II describes theoretical studies of ant colony optimization. Section III surveys the most successful ACO Variants. Section IV summarizes most notable applications. Section V presents conclusion for this study on ant colony optimization algorithms and highlights some current hot research topics.

I –BACKGROUND

The inspiring source of ACO algorithms are real ant colonies. Ants often find the shortest path between a food source and the nest of the colony without using visual information. In order to exchange information about which path should be followed, ants communicate with each other by means of a chemical substance called pheromone. As ants move, a certain amount of pheromone is dropped on the ground, creating a pheromone trail. The more ants follow a given trail, the more attractive that trail becomes to be followed by other ants. This process involves a loop of positive feedback, in which the probability that an ant chooses a path is proportional to the number of ants that have already passed by that path. Hence, individual ants, following very simple rules, interact to produce an intelligent behaviour at the higher level of the ant colony.

This shortest-path-finding process of the ant colony can be viewed as a form of swarm intelligence (Dorigo and Stützle, 2004). This basic behavior is the basis for a cooperative interaction for emergence of shortest paths. Figure 1.1 shows a simple example of how real ants find a shortest path



a) t=0 b) t=1 c) t=2 d) t=4

Figure 1.1. Illustration of ant colony principle, how real ants find shortest path in their search for food.

Let H be home, F be food source, and A-B be an obstacle in the route. (a) at time t=0, ants choose left and right side paths uniformly in their search for food source. (b) at time t=1, ants which have chosen the path F-B-H reach the food source earlier and are retuning back to their home, whereas ants which have chosen path H-A-F are still halfway in their journey to the food source. (c) at time t=2, since ants move at approximately constant speed, the ants which chose the shorter, right side path (H-B-F) reach the home faster, depositing more pheromone in H-B-F route. (d) at time t=4, pheromone accumulates at a higher rate on the shorter path (H-B-F), which is therefore automatically preferred by the ants and consequently all ants will follow the shortest path.The darkness of shade is approximately proportional to the amount of pheromone deposited by ants.

II- THE OPTIMIZATION TECHNIQUE.

Ant colony optimization (ACO) has been formalized into a metaheuristic for combinatorial optimization problems by Dorigo and co-workers [11] . A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems i.e. A metaheuristic is a general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications. (Examples of metaheuristics include simulated annealing , tabu search,iterated local search, evolutionary computation and ant colony optimization).This new metaheuristic has been shown to be both robust and versatile in the sense that it has been successfully applied to a range of different combinatorial optimization problems The design of an ACO algorithm involves following specification.

i-An appropriate representation of the problem, which allows the ants to incrementally construct/modify solutions through the use of a probabilistic transition rule, based on the amount of pheromone in the trail and problem-dependent heuristic.

ii-A method to enforce the construction of valid solutions, that is, solutions that are legal in the real-world situation corresponding to the problem definition.

iii-A problem-dependent heuristic function (η) that measures the quality of items that can be added to the current partial solution.

iv- A rule for pheromone updating, which specifies how to modify the pheromone trail (τ).

v-A probabilistic transition rule based on the value of the heuristic function (η) and on the contents of the pheromone trail (τ) that is used to iteratively construct a solution

The main rules of the ACO algorithm particulararly , ACS The most successful algorithm, are as follows.

Pseudo-random Proportional Rule

Ants use a decision rule called pseudo-random proportional rule, in which, an ant k in node i will select node j to move as follows

$$P^{k}_{ij}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{u \in J^k(i)} \{ [\tau_{iu}(t)]^{\alpha}[\eta_{iu}(t)]^{\beta} \}} & \text{if } j \in J^k(i) \\ 0 & \text{if } j \notin J^k(i) \end{cases} \quad \text{----(1)}$$

Where, $\eta_{ij}(t)$ represents heuristic information about the problem i.e. the heuristic value of Path i-j at time t according to the measure of the objective function; $\tau_{ij}(t)$ is the total Pheromone deposited on path i-j at time t ; $J^k(i)$ represents the allowable moves for ant k from node i; α and β are parameters that determine the relative importance of the Pheromone trail with respect to the heuristic information.

State Transition Rule

In each iteration of the algorithm, each ant progressively builds a solution, by using the probability transition rule. The next node j that ant k chooses to go to is given as,

$$j = \begin{cases} \max_{u \in J^k(i)} \{ [\tau_{iu}(t)]^{\alpha}[\eta_{iu}(t)]^{\beta} \} & \text{if } q \le q_0 \\ J & \text{if } q > q_0 \end{cases}$$

where q is a random number uniformly distributed in [0, 1]; q_0 is a tunable parameter ($0 \le q_0 \le 1$); $J \in J^k(i)$ is a node randomly selected according to the probability distribution given in Equation(1) .

Pheromone Updating Rules

Local Updating

The pheromone trail can be updated locally and globally. During construction of the solution, if an ant carries out the transition from node i to node j, then the pheromone value of the corresponding path will be changed as

step

$$\tau_{ij}(t) \leftarrow (1-\varphi).\tau_{ij}(t) + \varphi.\tau_0$$

where τ_0 is the initial value of pheromone; φ is a tunable parameter ($0 \le \varphi \le 1$).

Local updating is very useful to avoid premature convergence of the solution, and helps in exploring new search space, for the problems where the starting node is fixed.

Global Updating

At the end of an iteration of the algorithm, once all the ants have built a solution, pheromone trail is added to the arcs used by the ant that found the best tour from the beginning of the trail. The global trail updating rule is given as

iteration

$$\tau_{ij}(t) \leftarrow (1-\rho).\ \tau_{ij}(t) + \rho.\Delta\tau_{ij}$$

where ρ [0, 1] is a persistence parameter that controls the pheromone decay;

A pseudo-code for the ant colony system algorithm is given in Figure 1.2

Begin
Initialize
While stopping criterion not satisfied **do**
Position each ant in a starting node
Repeat
For each ant **do**
Choose next node by applying the state transition rule
Apply local pheromone update
End for
Until every ant has built a solution
Update best solution
Apply global pheromone update
End While
End

III- ACO Variants.

Several ACO algorithms have been proposed in the literature. The original ant colony optimization algorithm is known as Ant System[1] and was proposed in the early nineties. Since then, a number of other ACO algorithms were introduced. See Table I for a non-exhaustive list of successful variants. All ant colony optimization algorithms share the same characteristic idea and they generally differ from each other in the pheromone update rule that is applied. Here we present the original Ant System, and the two most successful variants: Ant Colony System and *MAX-MIN* Ant System.

TABLE-1 :

A non-exhaustive list of successful ant colony optimization algorithms (in chronological order)

Algorithms	Authors	Year
Ant system(AS)	Dorigo et al	1991
Elitist AS	Dorigo et al	1992
Ant-Q	Gambardella & Dorigo	1995
Ant Colony System	Dorigo & Gambardella	1996
MAX-MIN AS	Stutzle & Hoos	1996
Rank based AS	Bullnheimer et al.	1997
ANTS	Maniezzo	1999
BWAS	Cord´on et al.	2000
Hyper-cube AS	Blum et al.	2001

1) Ant System (AS): It is the first ACO algorithm proposed in the literature[1] . Its main characteristic is that, at each iteration, the pheromone values are updated by all the m ants that have built a solution in the iteration itself. The pheromone τ_{ij} , associated with the edge joining cities i and j, is updated as follows:

$$\tau_{ij} \leftarrow (1-\rho).\ \tau_{ij} + \sum_{k=1}^{m} \Delta\tau^{k}_{ij}$$

where ρ is the evaporation rate, m is the number of ants and $\Delta\tau^{k}_{ij}$ is the quantity of pheromone laid on edge (i, j) by ant k:

$$.\Delta\tau_{ij} = \begin{cases} Q / L_k & \text{if ant k used edge (i, j) in its tour,} \\ 0 & \text{otherwise,} \end{cases}$$

Where Q is a constant and L_k is the length of the tour constructed by ant k.

In the construction of a solution, ants select the following city to be visited through a stochastic mechanism. When ant k is in city i and has so far constructed the partial solution S^p, the probability of going to city j is given by

$$P^{k}_{ij} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\ \eta_{ij}^{\beta}}{\sum_{C_{iu} \in N(S^p)} \tau_{iu}^{\alpha}\ \eta_{iu}^{\beta}} & \text{if } C_{iu} \in N(S^p) \\ 0 & \text{otherwise} \end{cases}$$

where $N(S^p)$ is the set of feasible components; that is, edges (i, u) where u is a city not yet visited by the ant k. The parameters α and β control the relative importance of the pheromone versus the heuristic information η_{ij} , which is given by:

$$\eta_{ij} = 1 / d_{ij}$$

where d_{ij} is the distance between cities i and j.

2) Ant Colony System (ACS): The most interesting contribution of ACS is the introduction of a local pheromone update in addition to the pheromone update performed at the end of the construction process (called offline pheromone update). The local pheromone update is performed by all the ants after each construction step. Each ant applies it only to the last edge traversed:

$$\tau_{ij} \leftarrow (1-\varphi).\tau_{ij} + \varphi.\tau_0$$

where $\varphi \in (0, 1]$ is the pheromone decay coefficient, and τ_0 is the initial value of the pheromone. The main goal of the local update is to diversify the search performed by subsequent ants during an iteration: by decreasing the pheromone concentration on the traversed edges, ants encourage subsequent ants to choose other edges and, hence, to produce different solutions. This makes it less likely that several ants produce identical solutions during one iteration. The offline pheromone update, is applied at the end of each iteration by only one ant, which can be either the iteration-best or the best-so-far. However, the update formula is slightly different given by:

$$\tau_{ij} \leftarrow \begin{cases} (1-\rho).\tau_{ij} + \rho.\Delta\tau_{ij} & \text{if (i,j) belongs to best tour} \\ \tau_{ij} & \text{otherwise} \end{cases}$$

where $\Delta\tau_{ij} = 1 / L_{best}$, L_{best} is the length of the tour of the best ant where L_{best} can be either L_{ib} or L_{bs}.
L_{ib}- iteration best length, L_{bs} –best so far length.

The important difference between ACS and AS is in the decision rule used by the ants during the construction process. In ACS a new state transition rule called pseudo-random-proportional is introduced. The pseudorandom-proportional rule is a compromise between the pseudo-random state choice rule typically used in Q-learning and the random-proportional action choice rule typically used in Ant System. With the pseudo-random rule the chosen state is the best with probability q_0 (exploitation) while a random state is chosen with probability $1-q_0$ (exploration). Using the AS random-proportional rule the next state is chosen randomly with a probability distribution depending on τ_{ij} and η_{ij} . The ACS pseudo-random-proportional state transition rule provides a direct way to balance between exploration of new states and exploitation of a priori and accumulated knowledge. The best state is chosen with probability q_0 (that is a parameter $0 \le q_0 \le 1$ usually fixed to 0.9) and with probability $(1-q_0)$ the next state is chosen randomly with a probability distribution based on τ_{ij} and η_{ij} weighted by α (usually equal to 1) and β (usually equal to 2) .In ACS, the so-called pseudorandom proportional rule is used .the probability for an ant to move from city i to city j depends on a random variable q uniformly distributed over [0, 1], and a parameter q_0.

3) MAX-MIN Ant System (MMAS):

This algorithm is one of the best performing extension of original Ant System. Its characterizing elements are that only the best ant updates the pheromone trails and that the value of the pheromone is bound. The pheromone update is implemented as follows:

$$\tau_{ij}(t) \leftarrow [\,(1-\rho).\,\tau_{ij}(t) + \Delta\tau_{ij}^{\,best}]$$

where

$$\Delta\tau_{ij}^{\,best} = \begin{cases} 1 / L_{best} & \text{if (i, j) belongs to the best tour} \\ 0 & \text{otherwise} \end{cases}$$

L_{best} is the length of the tour of the best ant. This may be (subject to the algorithm designer decision) either the best tour found in the current iteration, iteration-best, L_{ib} or the best solution found since the start of the algorithm—best-so-far,L_{bs}—or a combination of both. Depending on some convergence measure,at each iteration either the IB-update or the BS-update rule are used for updating the pheromone values. At the start of the algorithm the IB-update rule is used more often, while during the run of the algorithm the frequency with which the BS-update rule is used increases.

IV. APPLICATIONS OF ANT COLONY OPTIMIZATION :

In recent years, the interest of the scientific community in ACO has risen sharply. In fact, several successful applications of ACO to a wide range of different discrete optimization problems are now available. The large majority of these applications are to NP-hard problems; that is, to problems for which the best known algorithms that guarantee to identify an optimal solution have exponential time worst case complexity. The use of such algorithms is often infeasible in practice and ACO algorithms can be useful for quickly finding high quality solutions. These applications are summarized in Table-2.

TABLE 2. Current Applications of ACO algorithms listed according to problem types

----

Problem Type & Name	**Main References**

Routing

Traveling Salesman	Dorigo et al. 1991a, Dorigo et al. 1991b, Dorigo et al. 1996, Gambardella and Dorigo 1995, Stutzle and Hoos 1997, Stutzle an Hoos 2000, Randall 2002, Tsai et al. 2004
Vehicle Routing	Bell and McMullen 2004, Bullnheimer et al. 1999a, Bullnheimer et al. 1999b, Gambardella et al. 1999, Reimann et al. 2002,
Sequential Ordering	Gambardella and Dorigo 1997, Gambardella and Dorigo 2000.

Assignment

Quadratic Assignment	Maniezzo 1999, Maniezzo and Colorni 1999, Maniezzo et al. 1994, Solimanpur et al. 2004, Solimanpur et al. 2005, Stutzle 1997, Stutzle and Hoos 2000, Talbi et al. 2001.
Graph	Costa and Hertz, 1997, Gamez and Puerta 2002, Gosavi et al. 2003, Korosec et al. 2004, Li and Wu 2003, Shyu et al. 2005.
Generalized Assignment	Eggers et al. 2003, Gutjahr 2002, Hoshyar et al. 2000, Lee Z.J. et al. 2002, Lourenco and Serra 1998, Lourenco and Serra 2002, Song et al. 1999, Stutzle and Hoos 2000, Xia et al. 2003, Yin 2003.
Frequency Assignment	Maniezzo and Carbonaro 2000, Lim et al. 2005.
University Course Timetabling	Azimi 2005, Socha et al. 2002, Socha et al. 2003.

Scheduling

Job shop	Color ni et al. 1994, Lee Z.J. and Lee 2004, Sendova-Franks and Lentz 2002.
Open shop	Li and Wu 2002, Pfahringer 1996, Ying and Liao 2004.
Flow shop	Gajpal and Rajendran 2005, Shyu et al. 2004, Stutzle 1998, Tkindt et al. 2002.
Total Tardiness	Bauer et al. 2000
Total weighted tardiness	den Besten et al. 2000, Gagne et al. 2002, Merkle and Middendorf 2000, Merkle and Middendorf 2003
Project Scheduling	Annaluru et al. 2004, Jayaraman et al. 2000, Merkleet al. 2000, Merkle et al. 2002, Samrout et al. 2005.
Group shop	Blum 2002, Blum 2003a

Subset

Multiple Knapsack	Leguizamon and Michalewicz 1999.Max Independent set Leguizamon and Michalewicz 2000.
Redundancy Allocation	Liang and Smith 1999, Zhao et al. 2005.
Set Covering	Hadji et al. 2000
Weighted Constraint Graph tree Partition	cordone and Maffioli 2001, Reimann and Laumanns 2005
Arc-weighted l-cardinality tree	Bullnheimer et al. 1999b
Maximum Clique	Fenet and Solnon 2003

Machine Learning

Classification Rules	Parpinelli et al. 2002, Vijayakumar et al. 2003, Zheng et al. 2003.
Bayesian Networks	de Campos et al. 2002
Fuzzy Systems	Bullnheimer et al. 1999c, Kuo et al. 2005

Network Routing

Connection-oriented Network routing	Bean and Costa 2005, Bonabeau et al. 1998, Di Caro and Dorigo 1998b,Schoonderwoerd et al. 1997, Su et al. 2005, White et al. 1998.
Connectionless Network	

Routing	Di Caro and Dorigo 1997, Di Caro and Dorigo 1998a, Di Caro and Dorigo 1998c, Heusse et al. 1998, Subramanian et al. 1997, Van der put 1998.
Optical Network Routing	Varela and Sinclair 1999.

Image Processing

Detection of Microcalcification in Digital Mammograms	Thangavel et al.2005b

V -Conclusion

Today, several hundred papers have been written on the applications of ACO. It is a true metaheuristic, with dozens of application areas. While both the performance of ACO algorithms and theoretical understanding of their working have significantly increased.There are several areas in which until now only preliminary steps have been taken and where much more research will have to be done. One of these research areas in the extension of ACO algorithms to more complex optimization problems that include (1) dynamic problems, in which the instance data, such as objective function values, decision parameters, or constraints, may change while solving the problem; (2) stochastic problems, in which one has only probabilistic information about objective function values, decision variables values, or constraint boundaries, due to uncertainty, noise, approximation, or other factors; and (3) multiple objective problems, in which a multiple objective function evaluates competing criteria of solution quality. Active research directions in ACO include also the effective parallelization of ACO algorithms and, on a more theoretical level, the understanding and characterization of the behavior of the ACO algorithms while solving a problem.

References:

[1] ***M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics – Part B, vol. 26, no. 1, pp. 29–41, 1996.***

[2] ***L. M. Gambardella and M. Dorigo, "Ant-Q: A reinforcement learning approach to the traveling salesman problem," in Proceedings of the Twelfth International Conference on Machine Learning (ML-95), A. Prieditis and S. Russell, Eds. Morgan Kaufmann Publishers, 1995, pp. 252–260.***

[3] ***M. Dorigo and L. M. Gambardella, "Ant colonies for the traveling salesman problem," BioSystems, vol. 43, no. 2, pp. 73–81, 1997.***

[4] ***"Ant Colony System": A cooperative learning approach to the traveling salesman problem," IEEE Transactions on Evolutionary Computation,vol. 1, no. 1, pp. 53–66, 1997.***

[5] ***L. M. Gambardella and M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies," in Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), T. Baeck et al., Eds. IEEE Press, Piscataway, NJ, 1996, pp. 622–627.***

[6] ***T. St¨utzle and H. H. Hoos, "MAX–MIN Ant System," Future Generation Computer Systems, vol. 16, no. 8, pp. 889–914, 2000. [16] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A new rank based version of the Ant System — a computational study," Institute of Management Science, University of Viena, Tech. Rep., 1997.***

[7] ***"A new rank-based version of the Ant System: A computational study," Central European Journal for Operations Research and Economics, vol. 7, no. 1, pp. 25–38, 1999.***

[8] ***V. Maniezzo, "Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem," INFORMS Journal on Computing, vol. 11, no. 4, pp. 358–369, 1999.***

[9] ***C. Blum, A. Roli, and M. Dorigo, "HC–ACO: The hyper-cube framework for Ant Colony Optimization," in Proceedings of MIC'2001 – Metaheuristics International Conference, vol. 2, Porto, Portugal, 2001, pp. 399–403.***

[10] ***C. Blum and M. Dorigo, "The hyper-cube framework for ant colony optimization," IEEE Transactions on Systems, Man, and Cybernetics – Part B, vol. 34, no. 2, pp. 1161–1172, 2004.***

[11] ***M. Dorigo and G. Di Caro, "The Ant Colony Optimization metaheuristic," in New Ideas in Optimization, D. Corne et al., Eds. McGraw Hill, London, UK, 1999, pp. 11–32.***

[12] ***M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," Artificial Life, vol. 5, no. 2, pp. 137–172, 1999.***

[13] ***M. Dorigo and T. St¨utzle, Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.***

[14] ***T. St¨utzle and H. H. Hoos, "The MAX–MIN Ant System and local search for the traveling salesman problem," in Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97), T. B¨ack et al., Eds. IEEE Press, Piscataway, NJ, 1997, pp. 309–314.***

[15] ***D. Merkle and M. Middendorf, "Modelling the dynamics of ant colony optimization algorithms," Evolutionary Computation, vol. 10, no. 3, pp. 235–262, 2002.***